CSE574 Introduction to Machine Learning
Jue Guo
Abstract
Introduction
Setup
Basic Formulation
Typical Scenario
Evaluation Metric
Theoretical Foundation and Analysis
Stability-Plasticity Trade-off

CSE574 Introduction to Machine Learning

Continual Learning

Jue Guo

University at Buffalo

January 30, 2024

Outline

1 Abstract

2 Introduction

3 Setup
- Basic Formulation
- Typical Scenario
- Evaluation Metric

4 Theoretical Foundation and Analysis
- Stability-Plasticity Trade-off

Abstract

To cope with real-world dynamics, an intelligent agent needs to incrementally acquire, update, accumulate, and exploit knowledge throughout its lifetime.

- **continual learning**, a fundation for AI systems to develop themselves adaptively. In a general sense, continual learning is explicitly limited by **catastrophic forgetting**, where learning a new task usually results in a dramatic performance degradation of the old tasks.

Introduction

Learning is the basis for intelligent systems to accommodate environments. In response to external changes, evolution has empowered human and other organisms with strong adaptability to continually acquire, update, accumulate and exploit knowledge. Naturally we expect artifical intelligence (AI) systems to adapt in a similar way.

- This motivates the study of **continual learning**, where a typical setting is to learn a sequence of contents one by one and behave as if they were observed simultaneously. Such contents could be new skills, new examples of old skills, different environments, different contexts, etc., with particular realistic challenges incorporated.
- As the contents are provided incrementally over a lifetime, continual learnming is also referred to as **incremental learning** or **lifelong learning** in much of the literature, without a strict distinction.

Unlike conventional machine learning models built on the premise of capturing a static data distribution, continual learning is characterized by learning from dynamic data distributions. A major challenge is known as **catastrophic forgetting**, where adaptation to a new distribution generally results in a largely reduced ability to capture old ones.

- This dilemma is a facet of the trade-off between **learning plasticity** and **memory stability**: an excess of the former interferes with the latter, and vice versa. A desirable solution for continual learning should obtain strong **generalizability** to accommodate distribution differences within and between tasks.
- A naive baseline, retraining all old training samples makes it easy to address the above challenges, but creates huge computational and storage overheads (as well as potential privacy issues).

In fact, continual learning is primarily intended to ensure the **resource efficiency** of model updates, perferably close to learning only new samples.

Numerous efforts have been devoted to addressing the above challenges, which can be conceptually seperated into five groups:

1. regularization-based approach
2. replay-based approach
3. optimization-based approach
4. representation-based approach
5. architecture-based approach

These methods are *closely connected*, e.g., regularization and replay ultimately act to rectify the gradient directions in optimization, and *highly synergistic*, e.g., the efficacy of replay can be facilitated by distilling knowledge from the old model.

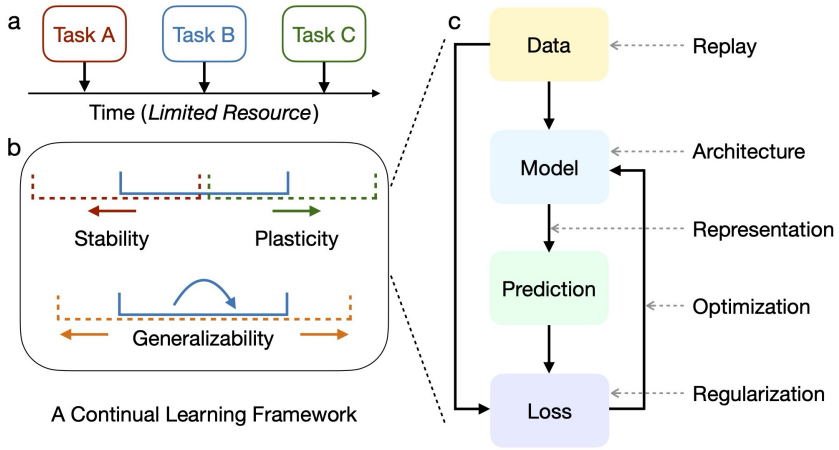

Figure: A conceptual framework of continual learning. **a**, Continual learning requires adapting to incremental tasks with dynamic data distributions. **b**, A desirable solution should ensure a proper balance between stability (red arrow) and plasticity (green arrow), as well as an adequate generalizability to intra-task (blue arrow) and inter-task (orange arrow) distribution differences. **c**, Representative strategies have targeted various aspects of machine learning.

Realistic applications present particular challenges for continual learning, which can be categorized into *scenario complexity* and *task specificity*.

- As for the former, for example, the task oracle (i.e., which task to perform) is probably missing in training and testing, and the training samples might be introduced in tiny batches or even one pass. Due to the expense and scarcity of data labeling, continual learning needs to be effective for few-shot, semi-supervised and even unsupervised scenarios.
- As for the latter, although current advances mainly focus on visual classification, other vision domains such as object detection, semantic segmentation and image generation, as well as other related fields, such as reinforcement learning (RL), natural language processing (NLP) and ethic considerations, are receiving increasing attention with their own opportunities and challenges.

Setup

- Basic formulation of continual learning
- Typica scenarios
- Evaluation metrics

Basic Formulation

Continual learning is characterized as learning from dynamic data distributions. In practice, training samples of different distributions arrive in sequence. A continual learning model parameterized by θ needs to learn corresponding task(s) with no or limited access to old training samples and perform well on their test sets.

- Formally, an incoming batch of training samples belonging to a task t can be represented as $\mathcal{D}_{t,b} = \{\mathcal{X}_{t,b}, \mathcal{Y}_{t,b}\}$, where $\mathcal{X}_{t,b}$ is the input data, $\mathcal{Y}_{t,b}$ is the data label, $t \in \mathcal{T} = \{1, \cdots, k\}$ is the task identity and $b \in \mathcal{B}_t$ is the batch index ($\mathcal{T}$ and $\mathcal{B}_t$ denote their space, respectively).
- Here we define a "task" by its training samples $\mathcal{D}_t$ following the distribution $\mathbb{D}_t := p(\mathcal{X}_t, \mathcal{Y}_t)$ ($\mathcal{D}_t$ denotes the entire training set by omitting the batch index, likewise for $\mathcal{X}_t$ and $\mathcal{Y}_t$), and assume that there is no difference in distribution between training and testing. Under realistic constraints, the data label $\mathcal{Y}_t$ and the task identity t might not be always available.

In continual learning, the training samples of each task can arrive incrementally in batches (i.e., $\left\{\{\mathcal{D}_{t,b}\}_{b\in\mathcal{B}_t}\right\}_{t\in\mathcal{T}}$) or simultaneously (i.e., $\{\mathcal{D}_t\}_{t\in\mathcal{T}}$).

Typical Scenario

According to the division of incremental batches and the availability of task identities, we detail the typical continual learning scenarios as follows:

- *Instance-Incremental Learning* (IIL): All training samples belong to the same task and arrive in batches.
- *Domain-Incremental Learning* (DIL): Tasks have the same data label space but different input distributions. Task identities are not required.
- *Task-Incremental Learning* (TIL): Tasks have disjoint data label spaces. Task identities are provided in both training and testing.
- *Class-Incremental Learning* (CIL): Tasks have disjoint data label spaces. Task identities are only provided in training.
- *Task-Free Continual Learning* (TFCL): Tasks have disjoint data label spaces. Task identities are not provided in either training or testing.
- *Online Continual Learning* (OCL): Tasks have disjoint data label spaces. Training samples for each task arrive as a one-pass data stream.
- *Blurred Boundary Continual Learning* (BBCL): Task boundaries are blurred, characterized by distinct but overlapping data label spaces.
- *Continual Pre-training* (CPT): Pre-training data arrives in sequence. The goal is to improve the performance of learning downstream tasks.

Scenario	Training	Testing
IIL [283]	$\{\{\mathcal{D}_{t,b}, t\}_{b\in\mathcal{B}_t}\}_{t=j}$	$\{p(\mathcal{X}_t)\}_{t=j}$; t is not required
DIL [169], [425]	$\{\mathcal{D}_t, t\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i = \mathcal{Y}_j$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$, t is not required
TIL [169], [425]	$\{\mathcal{D}_t, t\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is available
CIL [169], [425]	$\{\mathcal{D}_t, t\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is unavailable
TFCL [15]	$\{\{\mathcal{D}_{t,b}\}_{b\in\mathcal{B}_t}\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is optionally available
OCL [16]	$\{\{\mathcal{D}_{t,b}\}_{b\in\mathcal{B}_t}\}_{t\in\mathcal{T}}$, $\lvert b\rvert = 1$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is optionally available
BBCL [27], [47]	$\{\mathcal{D}_t, t\}_{t\in\mathcal{T}}$; $p(\mathcal{X}_i) \neq p(\mathcal{X}_j)$, $\mathcal{Y}_i \neq \mathcal{Y}_j$ and $\mathcal{Y}_i \cap \mathcal{Y}_j \neq \emptyset$ for $i \neq j$	$\{p(\mathcal{X}_t)\}_{t\in\mathcal{T}}$; t is unavailable
CPT [411]	$\{\mathcal{D}_t^{pt}, t\}_{t\in\mathcal{T}^{pt}}$, followed by a downstream task j	$\{p(\mathcal{X}_t)\}_{t=j}$; t is not required

Figure: A formal comparison of typical continual learning scenarios. $\mathcal{D}_{t,b}$: the training samples of task t and batch b.$|b|$: the size of batch b.$\mathcal{B}_t$: the space of incremental batches belonging to task t · $\mathcal{D}_t$: the training set of task t (further specified as $\mathcal{D}_t^{pt}$ for pre-training). $\mathcal{T}$: the space of all incremental tasks (further specified as $\mathcal{T}^{pt}$ for pre-training). $\mathcal{X}_t$: the input data in $\mathcal{D}_t$.$p(\mathcal{X}_t)$: the distribution of $\mathcal{X}_t$.$\mathcal{Y}_t$: the data label of $\mathcal{X}_t$.

If not specified, each task is generally assumed to have a sufficient number of labeled training samples, i.e., Supervised Continual Learning.

- According to the provided $\mathcal{X}_t$ and $\mathcal{Y}_t$ in each $\mathcal{D}_t$, continual learning can be further categorized into zero-shot, few-shot, semi-supervised, open-world (i.e., to identify unknown classes and then incorporate their labels) and un-/self-supervised scenarios.

Besides, other practical challenges have been considered and incorporated,

- such as multiple labels , noisy labels, hierarchical granularity and sub-populations , mixture of task similarity , long-tailed distribution , domain alignment , domain shifting, anytime inference, novel class discovery, multi-modality, etc.

Some recent work has focused on various combinations of these scenarios, as a way of better simulating the complexity of the real world.

Evaluation Metric

In general, the performance of continual learning can be evaluated from three aspects: overall performance of the tasks learned so far, memory stability of old tasks, and learning plasticity of new tasks.

- Here we summarize the common evaluation metrics, using classification as an example. **Overall performance, Memory stability**, and **Learning plasticity**.

Overall Performance

Overall performance is typically evaluated by *average accuracy* (AA) and *average incremental accuracy* (AIA).

- Let $a_{k,j} \in [0, 1]$ denote the classification accuracy evaluated on the test set of the j-th task after incremental learning of the k-th task ($j \leq k$). The output space to compute $a_{k,j}$ consists of the classes in either $\mathcal{Y}_j$ or $\cup_{i=1}^{k} \mathcal{Y}_i$, corresponding to the use of multi-head evaluation (e.g., TIL) or single-head evaluation (e.g., CIL).

The two metrics at the k-th task are then defined as

$$\mathrm{AA}_k = \frac{1}{k} \sum_{j=1}^{k} a_{k,j}$$

$$\mathrm{AIA}_k = \frac{1}{k} \sum_{i=1}^{k} \mathrm{AA}_i$$

where AA represents the overall performance at the current moment and AIA further reflects the historical variation.

Memory Stability

Memory stability can be evaluated by *forgetting measure* (FM) and *backward transfer* (BWT). As for the former, the forgetting of a task is calculated by the difference between its maximum performance obtained in the past and its current performance:

$$f_{j,k} = \max_{i \in \{1,\ldots,k-1\}} \left(a_{i,j} - a_{k,j}\right), \forall j < k.$$

FM at the k-th task is the average forgetting of all old tasks:

$$\mathrm{FM}_k = \frac{1}{k-1} \sum_{j=1}^{k-1} f_{j,k}$$

As for the latter, BWT evaluates the average influence of learning the k-th task on all old tasks:

$$\mathrm{BWT}_k = \frac{1}{k-1} \sum_{j=1}^{k-1} \left(a_{k,j} - a_{j,j}\right)$$

where the forgetting is usually reflected as a negative BWT.

Learning Plasticity

Learning plasticity can be evaluated by *intransience measure* (IM) and *forward transfer* (FWT). IM is defined as the inability of a model to learn new tasks, calculated by the difference of a task between its joint training performance and continual learning performance:

$$\mathrm{IM}_k = a_k^* - a_{k,k}$$

where a_k^* is the classification accuracy of a randomlyinitialized reference model jointly trained with $\cup_{j=1}^{k} \mathcal{D}_j$ for the k-th task.

In comparison, FWT evaluates the average influence of all old tasks on the current k-th task:

$$\mathrm{FWT}_k = \frac{1}{k-1} \sum_{j=2}^{k} (a_{j,j} - \tilde{a}_j)$$

where $\tilde{a}_j$ is the classification accuracy of a randomlyinitialized reference model trained with $\mathcal{D}_j$ for the j-th task. Note that, $a_{k,j}$ can be adapted to other forms depending on the task type, such as average precision (AP) for object detection, Intersection-over-Union (IoU) for semantic segmentation, Fréchet Inception Distance (FID) for image generation, normalized reward for reinforcement learning, etc, and the above evaluation metrics should be adjusted accordingly.

Theoretical Foundation and Analysis

Theoretical efforts in continual learning, with respect to stability-plasticity trade-off and generalizability analysis, and relate them to the main motivation for representative strategies.

Stability-Plasticity Trade-off

Let's consider a general setup for continual learning, where a neural network with parameters $\theta \in \mathbb{R}^{|\theta|}$ needs to learn k incremental tasks.

- The training set and test set of each task are assumed to follow the same distribution $\mathbb{D}_t$, $t = 1, \ldots, k$, where the training set $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{Y}_t\} = \{(x_{t,n}, y_{t,n})\}_{n=1}^{N_t}$ includes N_t data-label pairs.

The objective is to learn a probabilistic model $p(\mathcal{D}_{1:k} \mid \theta) = \prod_{t=1}^{k} p(\mathcal{D}_t \mid \theta)$ (with assumption of conditional independence) that can perform well on all tasks denoted as $\mathcal{D}_{1:k} := \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$.

- The task-related performance for discriminative models can be expressed as $\log p(\mathcal{D}_t \mid \theta) = \sum_{n=1}^{N_t} \log p_\theta(y_{t,n} \mid x_{t,n})$.

The central challenge of continual learning generally arises from the sequential nature of learning: when learning the k-th task from $\mathcal{D}_k$, the old training sets $\{\mathcal{D}_1, \ldots, \mathcal{D}_{k-1}\}$ are inaccessible. Therefore, it is critical but extremely challenging to capture the distributions of both old and new tasks in a balanced manner, i.e., ensuring an appropriate **stability-plasticity trade-off**, where excessive learning plasticity or memory stability can largely compromise each other.

A straightforward idea is to approximate and recover the old data distributions by storing a few old training samples or training a generative model, known as the replay-based approach.

- According to the learning theory for supervised learning, the performance of an old task is improved with replaying more old training samples that approximate its distribution, but with potential privacy issues and a linear increase in **resource overhead**.
- The use of generative models is also limited by the huge resource overhead, as well as their own catastrophic forgetting and expressiveness.

An alternative choice is to propagate the old data distributions in updating parameters through formulating continual learning in a *Bayesian framework*.

- Based on a prior, $p(\theta)$ the posterior after observing the k-th task can be computed with Bayes' rule:

$$p\left(\theta \mid \mathcal{D}_{1:k}\right) \propto p(\theta) \prod_{t=1}^{k} p\left(\mathcal{D}_t \mid \theta\right) \propto p\left(\theta \mid \mathcal{D}_{1:k-1}\right) p\left(\mathcal{D}_k \mid \theta\right)$$

  where the posterior $p\left(\theta \mid \mathcal{D}_{1:k-1}\right)$ of the $(k-1)$-th task becomes the prior of the k-th task, and thus enables the new posterior $p\left(\theta \mid \mathcal{D}_{1:k}\right)$ to be computed with only the current training set $\mathcal{D}_k$.

However, as the posterior is generally intractable (except very special cases), a common option is to approximate it with $q_{k-1}(\theta) \approx p\left(\theta \mid \mathcal{D}_{1:k-1}\right)$, likewise for $q_k(\theta) \approx p\left(\theta \mid \mathcal{D}_{1:k}\right)$.

Laplace Approximation

In the following, we will introduce two widely-used approximation strategies:

The first is *online Laplace approximation*, which approximates $p(\theta \mid \mathcal{D}_{1:k-1})$ as a multivariate Gaussian with local gradient information.

- Specifically, we can parameterize $q_{k-1}(\theta)$ with ϕ_{k-1} and construct an approximate Gaussian posterior $q_{k-1}(\theta) := q(\theta; \phi_{k-1}) = \mathcal{N}\left(\theta; \mu_{k-1}, \Lambda_{k-1}^{-1}\right)$ through performing a second-order Taylor expansion around the mode $\mu_{k-1} \in \mathbb{R}^{|\theta|}$ of $p(\theta \mid \mathcal{D}_{1:k-1})$, where Λ_{k-1} denotes the precision matrix and $\phi_{k-1} = \{\mu_{k-1}, \Lambda_{k-1}\}$, likewise for $q(\theta; \phi_k)$, μ_k and Λ_k.

The posterior mode for learning the current k-th task can be computed as

$$\begin{aligned}
\mu_k &= \arg\max_{\theta} \log p(\theta \mid \mathcal{D}_{1:k}) \\
&\approx \arg\max_{\theta} \log p(\mathcal{D}_k \mid \theta) + \log q(\theta; \phi_{k-1}) \\
&= \arg\max_{\theta} \log p(\mathcal{D}_k \mid \theta) - \frac{1}{2}(\theta - \mu_{k-1})^{\top} \Lambda_{k-1}(\theta - \mu_{k-1}),
\end{aligned}$$

which can be obtained from μ_{k-1} and Λ_{k-1}.

Similarly, Λ_k can be updated recursively:

$$\begin{aligned}\Lambda_k &= -\nabla_\theta^2 \log p\left(\theta \mid \mathcal{D}_{1:k}\right)\Big|_{\theta=\mu_k} \\ &\approx -\nabla_\theta^2 \log p\left(\mathcal{D}_k \mid \theta\right)\Big|_{\theta=\mu_k} + \Lambda_{k-1}\end{aligned}$$

where the first term on the right side is the Hessian of the negative log likelihood of $\mathcal{D}_k$ at μ_k, denoted as $H\left(\mathcal{D}_k, \mu_k\right)$.

- In practice, $H\left(\mathcal{D}_k, \mu_k\right)$ is often computationally inefficient due to the great dimensionality of $\mathbb{R}^{|\theta|}$, and there is no guarantee that the approximated Λ_k is positive semi-definite for the Gaussian assumption.

To overcome these issues, the Hessian can be approximated by the Fisher information matrix (FIM):

$$F_k = \mathbb{E}\left[\nabla_\theta \log p\left(\mathcal{D}_k \mid \theta\right) \nabla_\theta \log p\left(\mathcal{D}_k \mid \theta\right)^\top\right]\Big|_{\theta=\mu_k} \approx H\left(\mathcal{D}_k, \mu_k\right)$$

The computation of FIM can be further simplified with a diagonal approximation or a Kronecker-factored approximation.

Formally,

$$\begin{aligned}\mu_k &= \arg\max_{\theta} \log p\left(\theta \mid \mathcal{D}_{1:k}\right) \\ &\approx \arg\max_{\theta} \log p\left(\mathcal{D}_k \mid \theta\right) + \log q\left(\theta; \phi_{k-1}\right) \\ &= \arg\max_{\theta} \log p\left(\mathcal{D}_k \mid \theta\right) - \frac{1}{2}\left(\theta - \mu_{k-1}\right)^{\top} \Lambda_{k-1}\left(\theta - \mu_{k-1}\right),\end{aligned}$$

is implemented by saving a copy of the old model μ_{k-1} to regularize parameter changes, which is known as the *regularization based approach*. Here, we use EWC as an example and present its loss function:

$$\mathcal{L}_{\mathrm{EWC}}(\theta) = \ell_k(\theta) + \frac{\lambda}{2}\left(\theta - \mu_{k-1}\right)^{\top} \hat{F}_{1:k-1}\left(\theta - \mu_{k-1}\right),$$

where ℓ_k denotes the task-specific loss, the FIM $\hat{F}_{1:k-1} = \sum_{t=1}^{k-1} \mathrm{diag}\left(F_t\right)$ with a diagonal approximation $\mathrm{diag}(\cdot)$ of each F_t, and λ is a hyperparameter to control the strength of regularization.

Variational Inference

The second is *online variational inference* (VI). There are many different ways to do this, and a representative one is to minimize the following KL-divergence over a family $\mathcal{Q}$ that satisfies $p\left(\theta \mid \mathcal{D}_{1:k}\right) \in \mathcal{Q}$ at the current k-th task: